



ROBOFUN.RO

LECȚIA X

CURS GRATUIT

ARDUINO ȘI ROBOTICĂ

LCD-uri pentru Arduino

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs
CC BY-NC-ND



Codul sursa din acest document este licentiat

Public-Domain

Esti liber sa distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, printare, sau orice alt mijloc), atat timp cat nu aduci nici un fel de modificari acestuia. Codul sursa din acest document poate fi utilizat in orice fel de scop, de natura comerciala sau nu, fara nici un fel de limitari.

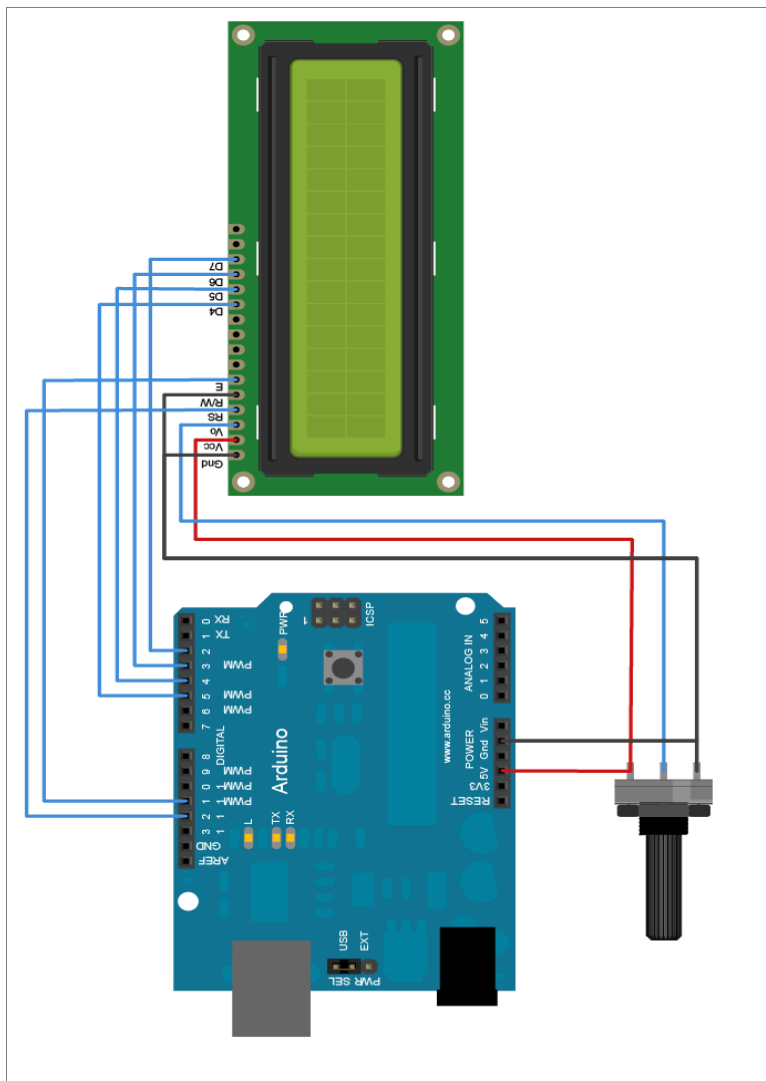
LCD 16 X 2, LCD 16 X 4, LCD 20 X 4

Toate aceste LCD-uri se interfateaza cu Arduino exact la fel. Singura diferenta este dimensiunea lor. Astfel, LCD-ul 16 X 2 permite 16 coloane si doua linii de caractere (adica in total 32 de caractere), LCD-ul 16X4 permite 4 linii (64 de caractere in total), iar cel de 20X4 permite 80 de caractere afisate.

Ca sa-l folosesti cu Arduino, ai nevoie de fire de conectare si de un potentiometru de 10K pentru reglarea contrastului (nu merge fara el). Schema de conectare este cea de mai jos (nu este simpla).

Daca esti la primele tale teste cu Arduino, din pacate s-ar putea sa gasesti conectarea LCD-ului o experienta frustranta. Sunt destul de multe fire de conectat, si daca gresesti unul singur nu-ti va functiona. Iti recomand sa faci montajul incet si sa verifici de cateva ori fiecare conexiune. Daca ai ajuns la final, vei avea atat de multe fire incat iti va fi dificil sa vezi care fir duce unde. Verifica cu atentie pe parcurs ca sa fii sigur ca fiecare fir duce acolo unde trebuie.

Daca totusi nu ti-a iesit si vrei sa incerci ceva mai simplu, iti recomand shield-ul cu LCD 2X16 (<http://www.robofun.ro/bricks/shield-lcd-16x2>) care nu face altceva decat sa-ti ascunda conexiunile complicate.



Arduino 5 V	Pin1 Potentiometru
Arduino GND	Pin3 Potentiometru
Pin2 Potentiometru	V0 (PIN3) LCD
Arduino GND	GND (PIN1) LCD
Arduino GND	RW (PIN5) LCD
Arduino 5 V	VCC (PIN2) LCD
Arduino Digital 12	RS (PIN4) LCD
Arduino Digital 11	E (PIN6) LCD
Arduino Digital 5	D4 (PIN11) LCD

Arduino Digital 4	D5 (PIN12) LCD
Arduino Digital 3	D6 (PIN13) LCD
Arduino Digital 2	D7 (PIN14) LCD

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
  lcd.begin(16, 2);
  lcd.print("hello, world!");
}

void loop() {
  lcd.setCursor(0, 1);
  lcd.print(millis()/1000);
}
```

Codul sursa de mai sus utilizeaza libraria LiquidCrystal (inclusa in mod standard in mediul de dezvoltare Arduino). Singura modificare pe care va trebui sa o faci cand folosesti un alt tip de LCD este sa schimbi parametrii din rutina de initializare. Astfel, pentru un LCD 20X4, linia 3 se schimba la "lcd.begin(20, 4)".

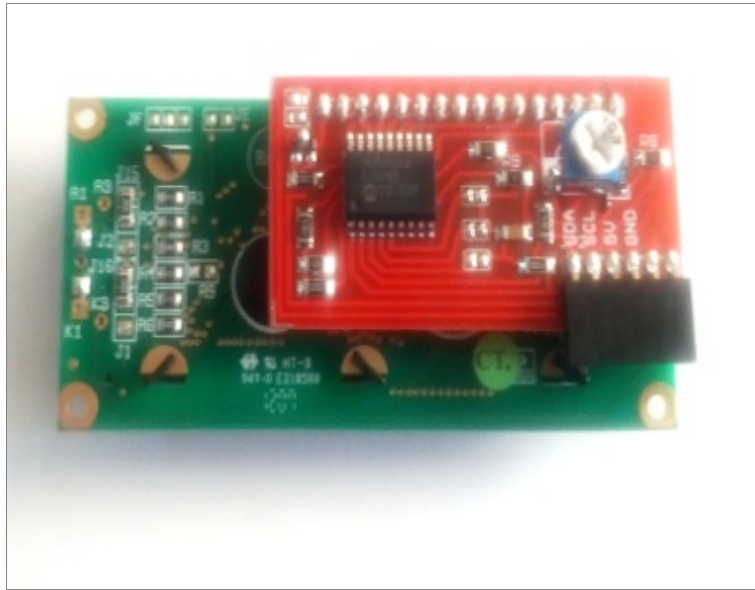
Alte rutine interesante ale libreriei LiquidCrystal sunt mai jos :

- **clear()** - curata ecranul complet. Se apeleaza fara parametri - "lcd.clear()"
- **home()** - muta cursorul in partea din stanga, sus. Se apeleaza fara parametri - "lcd.home()"
- **setCursor()** - muta cursorul la pozitia specificata. Textul care urmeaza a fi scris este scris la pozitia specificata de aceasta metoda. Astfel, pentru a scrie text pe linia 2, coloana 4, vom apela "lcd.setCursor(1,3); lcd.print("TEXT");"
- **noDisplay()** - opreste LCD-ul, fara a pierde textul afisat. Se apeleaza fara parametri - "lcd.noDisplay()"
- **display()** - porneste LCD-ul dupa ce a fost oprit folosind "noDisplay". Se apeleaza fara parametri - "lcd.display()"
- **scrollDisplayLeft()** - deplaseaza textul afisat pe LCD cu un caracter spre stanga. Se apeleaza fara parametri - "lcd.scrollDisplayLeft()"
- **scrollDisplayRight()** - deplaseaza textul afisat pe LCD cu un caracter spre dreapta. Se apeleaza fara parametri - "lcd.scrollDisplayRight()"

LCD 16 X 2 pe I2C, LCD 20 X 4 pe I2C

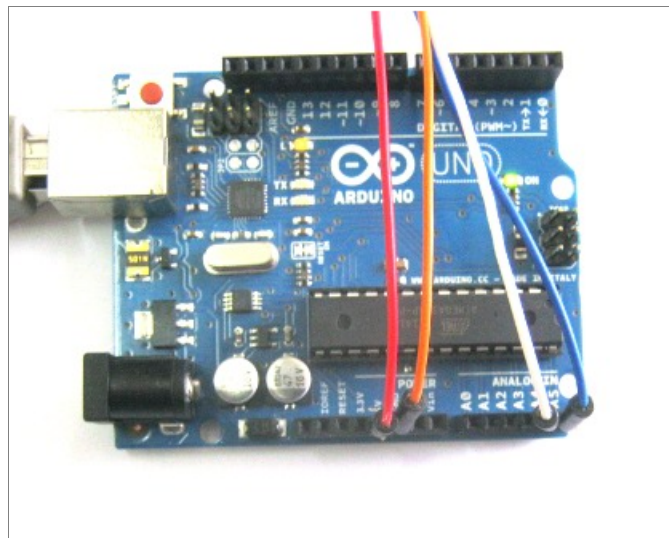
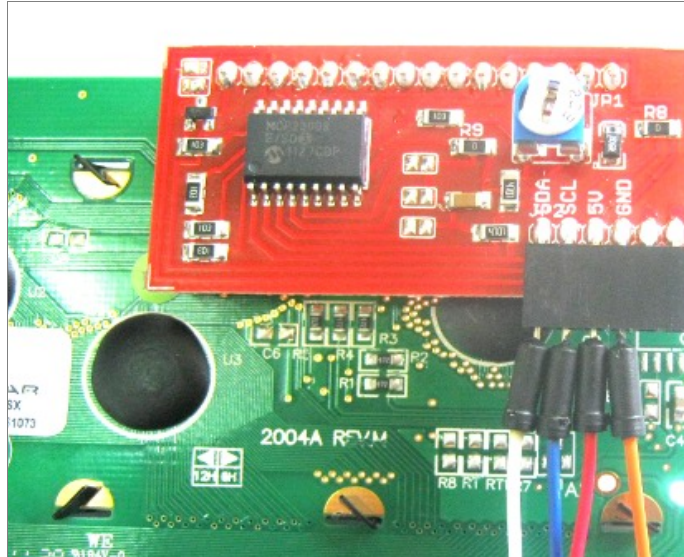
Ambele LCD-uri sunt LCD-urile obisnuite despre care am discutat mai sus, carora li s-a atasat o placa suplimentara care comunica pe I2C cu Arduino si seteaza cei 8 pini pentru controlul LCD-ului la valorile care trebuie astfel incat pe LCD sa fie afisat text-ul care trebuie. In acest fel, intre Arduino si LCD sunt necesare doar doua fire (SDA si SCL), in afara firului de GND si alimentare.





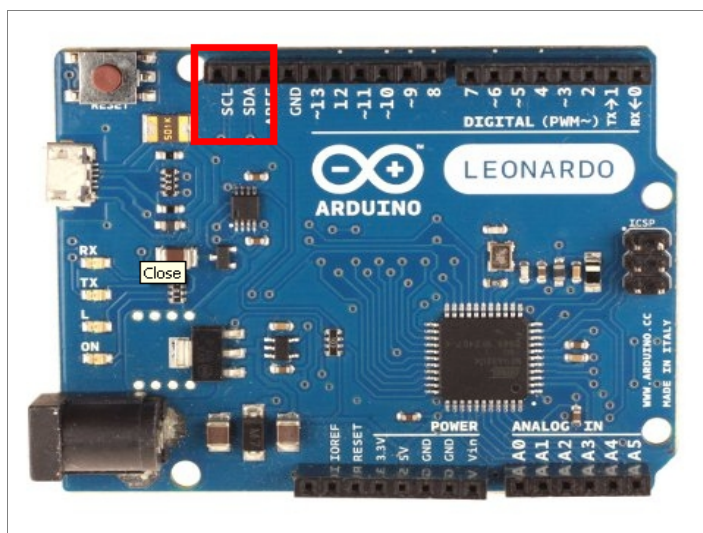
Conexiunile la Arduino sunt ca in cele doua imagini de mai jos. Ultimii doi pini din mufa lipita pe placa LCD-ului nu se folosesc. In rest, de la stanga la dreapta, avem SDA (se conecteaza la pinul analogic 4 pe Arduino UNO sau la pinul SDA pe Arduino Leonardo), SCL (se conecteaza la pinul analogic 5 pe Arduino UNO sau la pinul SCL pe Arduino Leonardo), 5V (se conecteaza la pinul 5V pe Arduino, si pinul GND (se conecteaza la pinul GND).

Arduino UNO



Arduino 5 V	LCD 5V
Arduino GND	LCD GND
Arduino Analog 4	LCD SDA
Arduino Analog 5	LCD SCL

Arduino Leonardo

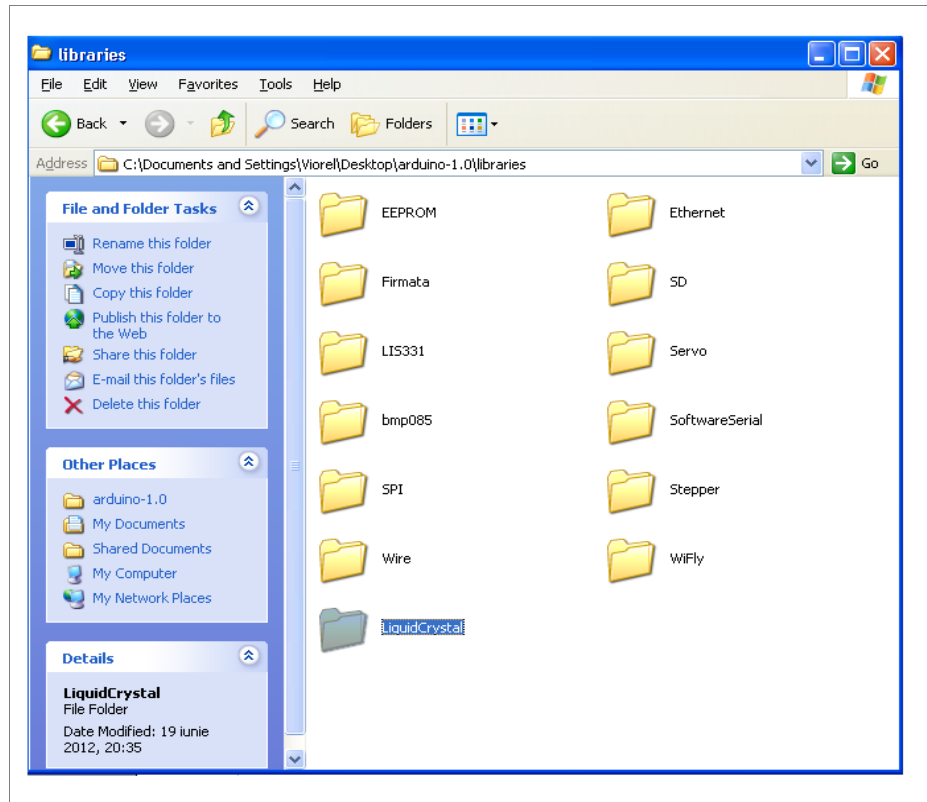


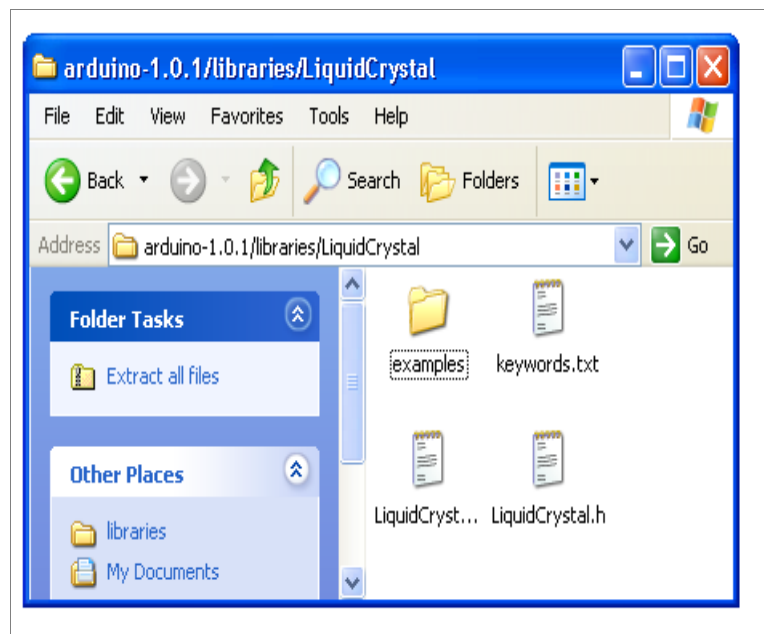
Arduino 5 V	LCD 5V
Arduino GND	LCD GND
Arduino SDA	LCD SDA
Arduino SCL	LCD SCL

Urmatorul pas este actualizarea librăriei LCD, astfel incat sa suporte si comunicarea I2C.

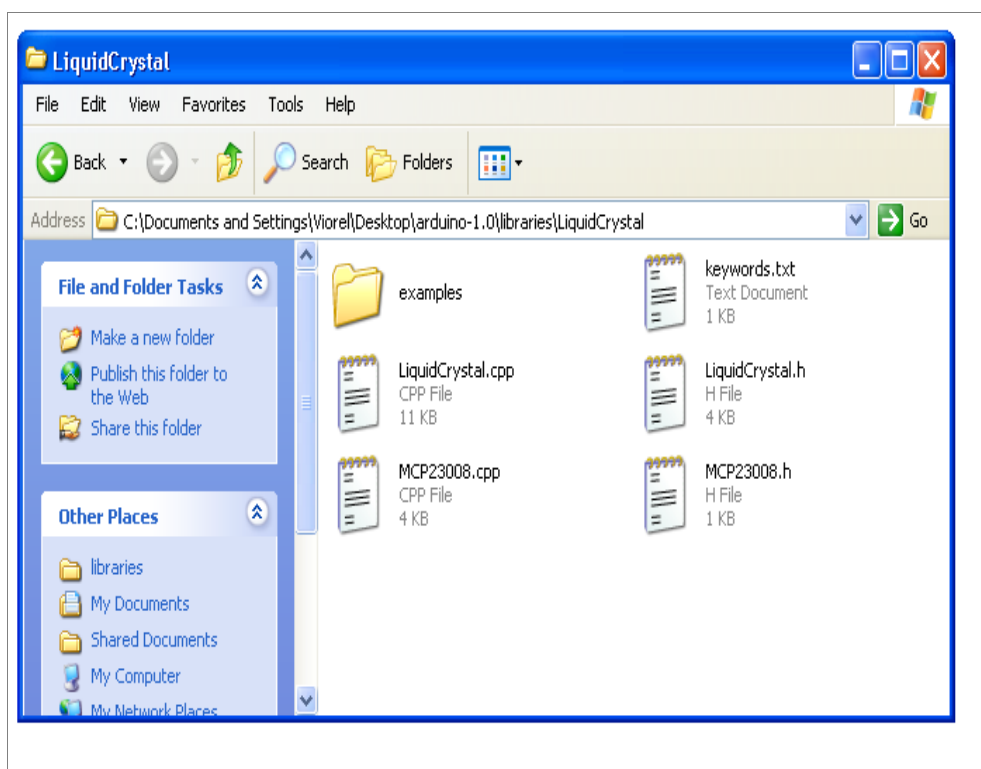
Mergi la adresa http://www.robofun.ro/lcd_20x4_i2c_negru_verde si descarca librăria din acea pagina. Este bine sa o incerci prima data pe cea de pe GITHUB, care este intotdeauna la zi. Dupa ce ai descarcat-o, inchide mediul Arduino si deschide folder-ul "libraries" din folder-ul in care este instalat mediul Arduino. Folder-ul "libraries" ar trebui sa arate similar cu imaginea de mai jos.

Deschide si folder-ul "LiquidCrystal" ca sa vezi ce fisiere sunt in interior. Ar trebui sa vezi ceva similar cu ce este mai jos.





Sterge apoi tot ceea ce este in acest folder si inlocuieste cu fisierele descarcate anterior. Ar trebui sa fie similar cu imaginea de mai jos.



Porneste acum mediul de dezvoltare Arduino si incearca programul de mai jos.

```
#include "Wire.h"
#include "LiquidCrystal.h"
LiquidCrystal lcd(0);
void setup() {
    lcd.begin(20, 4);

    lcd.setBacklight(HIGH);
    lcd.print("hello, world 0 !");
    lcd.setCursor(0, 1);
    lcd.print("hello, world 1 !");
    lcd.setCursor(0, 2);
    lcd.print("hello, world 2 !");
    lcd.setCursor(0, 3);
    lcd.print("hello, world 3 !");
}

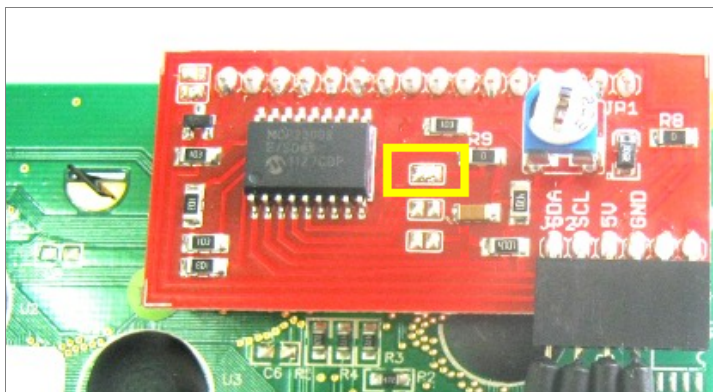
void loop() {
}
```



Daca LCD-ul tau nu arata ca mai sus, pe spatele placutei rosii vei gasi un potentiometru de culoare albastra. Acest potentiometru stabileste contrastul LCD-ului, si probabil ca s-a miscat in timpul transportului. Folosind o surubelnita mica, un varf de cutit ascutit, foarfeca sau pila de unghii, roteste-l usor si urmareste in acelasi timp textul pe LCD pana cand devine foarte clar.

Si partea frumoasa abia acum vine ! Cu acest tip de LCD, poti conecta simultan pana la opt LCD-uri la acelasi Arduino, folosind aceeasi doi pini I2C. Pentru aceasta, intoarce LCD-ul pe spate, si observa cei 3 jumperi pentru

setarea adresei I2C. In mod obisnuit, nici unul dintre acesti jumperi nu este lipit, asa ca adresa shield-ului este zero (lucru pe care il vezi in cod la linia *LiquidCrystal lcd(0)*). Fiecare LCD va trebuie sa aiba o adresa I2C diferita, asa ca ceea ce ai de facut este sa folosesti un letcon si putin fludor pentru a conecta unul sau mai multi jumperi impreuna. Adresele sunt in cod binar, astfel ca folosind cei trei jumperi poti obtine exact opt adrese. Pentru a conecta un jumper, incalzeste ambele pad-uri, apoi adauga fludor si intinde fludorul astfel incat sa faca contact intre ambele pad-uri, ca mai jos.



Pentru a conecta mai multe LCD-uri la acelasi Arduino, tot ce ai de facut este conectezi toti pinii de SDA impreuna de la toate LCD-urile, toti pinii SCL impreuna, si la fel si GND si 5V. In prealabil, ai grija sa setezi adrese I2C diferite lipind cei trei jumperi de adresa in configuratii diferite. Apoi, in cod definesti mai multe obiecte de tip LCD, ca mai jos.

```
#include "Wire.h"
#include "LiquidCrystal.h"

LiquidCrystal lcd1(0);
LiquidCrystal lcd2(1);
LiquidCrystal lcd3(2);

void setup() {
  lcd1.begin(20, 4);
  lcd2.begin(20, 4);
  lcd3.begin(20, 4);

  lcd1.setBacklight(HIGH);
  lcd1.print("LCD1, hello, world 0 !");
  lcd1.setCursor(0, 1);
  lcd1.print("LCD1, hello, world 1 !");
  lcd1.setCursor(0, 2);
  lcd1.print("LCD1, hello, world 2 !");
  lcd1.setCursor(0, 3);
  lcd1.print("LCD1, hello, world 3 !");

  lcd2.setBacklight(HIGH);
  lcd2.print("LCD2, hello, world 0 !");
```

```
lcd2.setCursor(0, 1);  
lcd2.print("LCD2, hello, world 1 !");  
lcd2.setCursor(0, 2);  
lcd2.print("LCD2, hello, world 2 !");  
lcd2.setCursor(0, 3);  
lcd2.print("LCD2, hello, world 3 !");  
  
lcd3.setBacklight(HIGH);  
lcd3.print("LCD3, hello, world 0 !");  
lcd3.setCursor(0, 1);  
lcd3.print("LCD3, hello, world 1 !");  
lcd3.setCursor(0, 2);  
lcd3.print("LCD3, hello, world 2 !");  
lcd3.setCursor(0, 3);  
lcd3.print("LCD3, hello, world 3 !");  
}  
  
void loop() {  
  
}
```

Aceasta a fost lectia 10. In final, as vrea sa te rog sa ne oferi feedback asupra acestei lectii, pentru a ne permite sa le facem mai bune pe urmatoarele.

Este vorba despre un sondaj cu 4 intrebari (oricare este optionala), pe care il poti accesa [dand click aici](#).

Sau ne poti contacta direct prin email la contact@robofun.ro .

Iti multumim,

Echipa [Robofun.RO](#)